

摘要：最近邻问题定义如下:给定n个点的集合P={p1,….pn}在某个度量空间X,为了高效地回答需要找到P集合中最接近查询点q的点(q∈X)的查询需要预处理P. 我们关注 d 维欧几里得空间的特别有趣的情况，其中在某些 lp 范数下 X = Rd。尽管经过了数十年的努力，当前的解决方案远远不能让人满意;事实上，对于很大的d,无论在理论上还是在实践中，它们相对于将查询点与每个数据点进行比较的蛮力算法几乎没有提供任何改进。最近，人们对近似最近邻问题有了兴趣，即:找到查询q的c近似最近邻点p∈P,对于所有p’∈P,d(p,q)≤(1+c)d(p’,q)

我们提出了两个近似版本的算法结果，可以显着改善已知界限：（a）n 和 d 中的预处理成本多项式，以及真正的次线性查询时间（对于 c>1）(b) logn 和 d 中的查询时间多项式，并且只有轻度指数预处理成本 O(n)\*O(1/c)d . 此外，在随机投影上应用经典的几何引理（对此我们给出了更简单的证明），我们获得了第一个已知的具有多项式预处理和查询时间多项式的算法 d 和 logn。不幸的是，对于较小的c, 后者纯粹是理论结果，因为指数取决于 1/c。试验结果表明我们的第一个算法在实际数据集上的运行时间上提供了数量级的改进。它的关键成分是**局部敏感哈希**的概念，这可能是独立的兴趣；在这里，我们给出了信息检索、模式识别、动态最近对和快速聚类算法的应用。

1介绍

最近邻搜索(NNS)问题是:给定某个度量空间X中n个点的集合P={p1,….pn}，(空间具有)距离函数 d，对 P 进行预处理，以便有效地回答查询，以找到 P 中最接近查询点 q∈X 的点. 我们关注 d 维欧几里得空间的特别有趣的情况，其中在某些 lp 范数下 X = Rd。低维情况已经被很好解决，因此主要的问题就是解决“维度诅咒”[16]。问题最初在1960s被Minsky 和Papert[53,pp.222-225]提出。管经过了数十年的努力，当前的解决方案远远不能让人满意。事实上，对于很大的d,无论在理论上还是在实践中，它们相对于将查询点与每个数据点进行比较的蛮力算法几乎没有提供任何改进。最近，人们对近似最近邻问题有了兴趣，即:找到查询q的c近似最近邻点p∈P,对于所有p’∈P,d(p,q)≤(1+c)d(p’,q)

我们提出了两个近似版本的算法结果，可以显着改善已知界限：（a）n 和d 中的预处理成本多项式，以及真正的次线性查询时间（对于 c>1）(b) logn 和 d 中的查询时间多项式，并且只有轻度指数预处理成本 O(n)\*O(1/c)d . 此外，在随机投影上应用经典的几何引理（对此我们给出了更简单的证明），我们获得了第一个已知的具有多项式预处理和查询时间多项式的算法 d 和 logn。不幸的是，对于较小的c, 后者纯粹是理论结果，因为指数取决于 1/c。试验结果表明我们的第一个算法在实际数据集上的运行时间上提供了数量级的改进。它的关键成分是**局部敏感哈希**的概念，这可能是独立的兴趣；在这里，我们给出了信息检索、模式识别、动态最近对和快速聚类算法的应用。

**动机** 最近邻问题对于各种应用都非常重要，通常涉及**相似性搜索。**一些示例是:数据压缩[35];数据库和数据挖掘[12,38];信息检索[10,20,57];图像和视频数据库[28,30,55,60];机器学习[18]；模式识别[19,25]；以及数据分析[21,44]。

通常，感兴趣的对象（文档、图像等）的特征表示为 Rd 中的点，并且使用**距离度量来测量对象的相似性**（不相似性）。那么基本问题是对查询对象执行索引或相似性搜索。特征的数目(也就是维数)范围从数十到千.例如，在IBM的QBIC等多媒体应用中，特征的数量可能达到数百个。在文本文档的信息检索中，向量空间表示涉及数千个维度，这被认为是降维的一个巨大改进。 诸如LSI（潜在语义索引）、主成分分析或Karhunen-Loeve变换等技术可以将维度减少到仅仅几百个。

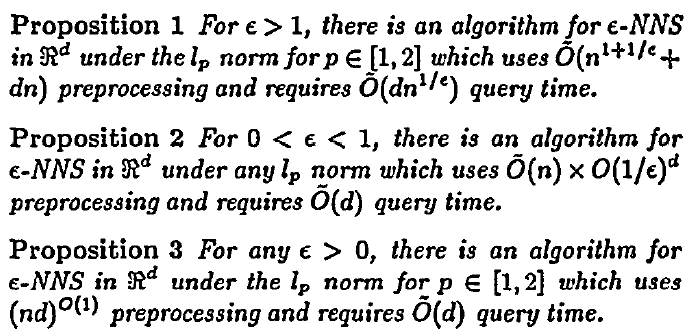
最近，对于避免维度灾难，人们越来越关注采用近似最近邻搜索的方法。由于应用程序中特征的选择和距离度量的使用相当启发式，并且仅仅是试图在数学上精确（毕竟本质上是相似性的美学概念），因此坚持绝对最近邻似乎是一种矫枉过正；实际上，确定一个合理值 c 的 c-近似最近邻，比如一个小常数，对于大多数实际目的来说应该已经足够。不幸的是，即使目标放宽了一些，也没有消除维度灾难，尽管 Kleinberg [45] 最近的结果有一些改进。

近期工作，Samet [58] 对各种用于最近邻搜索的数据结构进行了综述，包括 I;kd 树的变体、R 树以及基于空间填充曲线的结构；更近期的研究结果见 [59]。虽然其中一些在2-3维空间表现良好，但在高维空间中它们在最坏情况下和典型情况下都表现糟糕（例如 Arya、Mount 和 Narayan [3] 的情况）。Dobkin 和 Lipton [22] 首次提供了一个在Rd中搜索最近邻的算法，查询时间为O(2dlogn)预处理时间为O(n^2^(d+1))，Clarkson [15] 在减小预处理时间至的同时增加查询时间至。后续的结果，例如 Agarwal 和 Matoušek [1]、Matoudek [50] 以及 Yao 和 Yao [64]，都面临着在d方面指数级的查询时间。Meiser [51] 在O(nd+δ)的预处理后获得了查询时间为O(d^5\*log n)。所谓的“视角点”技术 [11, 12, 61, 62] 是一种近期流行的启发式方法，但我们对于在高维欧氏空间中的其分析并不了解。总的来说，即使是对于分布在Rd中的点的启发式方法的平均情况分析也会导致指数级的查询时间 [6, 34, 58]。

对于近似最近邻搜索，情况只略有改善。Arya 和 Mount [2] 提出了一个算法，查询时间为 O(l/c)dO(log n)，预处理为 O(l/e)dO(n)。Clarkson [16] 和 Ghan [14] 后来将对 c 的依赖降低为 c-td-r)^(1/2)。Arya、Mount、Netanyahu、Silverman 和 Wu [4] 获得了最优的 O(n) 预处理成本，但查询时间增长为 O(dd)。Bern [7] 和 Chan [14] 考虑了在 d 中多项式级别的误差 c，并设法在这种情况下避免指数依赖。最近，Kleinberg [45] 提出了一种算法，预处理为 o(n log d)^(2d)，查询时间为多项式级别的 d、c 和 log n；另一种算法在预处理中是 d、c 和 n 的多项式，但查询时间为 O(n + dlog^3 n)。后者改进了蛮力算法的 O(dn) 时间上限。

对于 Hamming 空间（{0,1}^d），Dolev、Harari 和 Parnas [24] 以及 Dolev、Harari、Liial、Nisan 和 Parnas [23] 提供了检索距离查询点 q 距离 r 内所有点的算法。然而，对于任意 P，这些算法在查询时间或预处理中都是指数级的。Greene、Parnas 和 Yao [37] 提出了一种方案，对于均匀随机选择的二进制数据，在时间 O(dn^(1/d)) 内检索距离查询点 q 距离 r 内的所有点，使用 O(dn^(1/d)) 的预处理。

最近，Kushilevitz、Ostrovsky 和 Rabani [46] 得到了类似于下面 命题 3 的结果。

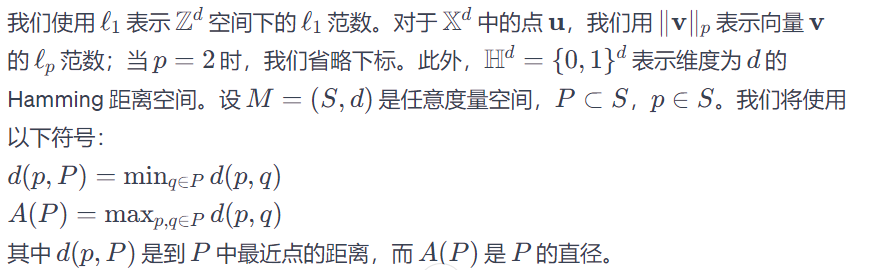


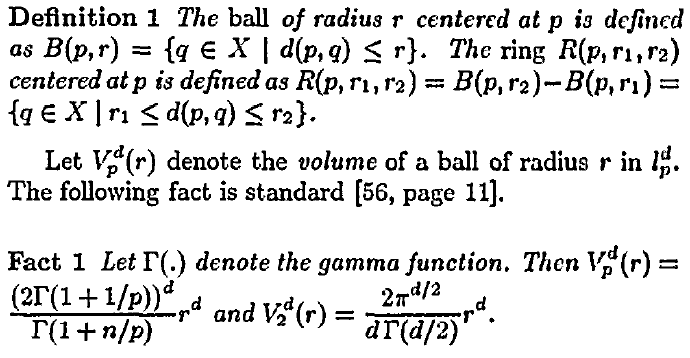
我们通过将最近邻搜索问题（c-NNS）归约为一个新问题，即**等球体中的点定位**，这些结果是通过一种称为“ring-cover trees”的新颖数据结构获得的，详见第3节。我们的技术可以看作是参数搜索 [52] 的一种变体，其中它们允许我们将优化问题归约为其决策版本。主要区别在于，在回答查询时，我们只能要求解决方案属于预定集合的决策问题，因为解决决策问题（即在等球体中定位点）需要在预处理期间创建的数据结构。我们相信这种技术将在参数搜索曾经有帮助的问题中找到进一步的应用。

在第4节，我们给出了两种解决点定位问题的方法。其中一种基于一种类似于Elias bucketing算法 [63] 的方法 - 我们将每个球体分解为有限数量的单元，并将它们存储在字典中。这使我们能够实现 O(d) 的查询时间，而预处理在 d 方面是指数级的，符合 命题 2。对于第二个解决方案，我们引入了**局部敏感哈希**的技术。关键思想是使用哈希函数，使得对象之间的碰撞概率对于彼此靠近的对象要高得多，而对于远离的对象要低得多。我们证明，对于任何域（不一定是度量空间），这种函数的存在意味着该域存在快速的 c-NNS 算法，其预处理成本仅与 d 线性相关，与 n 亚线性相关（对于 e > 1）。然后，我们提供了两类这种函数 - 一类用于 Hamming 空间，另一类用于 Broder 等人 [9] 用于聚类 Web 文档的相似性度量下的一组子集。基于第一类的算法用于通过以保持距离的方式将 Rd 中的点嵌入 Hamming 立方体来获得 Rd 中数据集的最近邻算法。基于相似性度量的算法显示出对信息检索和模式识别的多个应用。我们还为局部敏感哈希提供了动态最近对问题和快速聚类算法的其他应用。所有基于这种方法的算法都易于实现，并具有其他优势 - 它们利用数据的稀疏性，而在实践中的运行时间比理论分析所预测的要低得多 [36]。我们期望这些结果将在实践中产生显著的影响。

一种优雅的降低由于维度引起的复杂性的技术是将点投影到较低维度的随机子空间中，例如通过将 P 投影到通过原点的一小组随机直线。具体而言，我们可以利用 Frankl 和 Maehara [32] 的结果，该结果改进了 Johnson-Lindenstrauss 引理 [41]，表明将 P 投影到由大约  条随机直线定义的子空间，以很高的概率保持所有点间距的相对误差为 e。将这一结果应用于查询时间为 o(l)d 的算法，我们得到一个查询时间为 的算法。不幸的是，这只对较大的 E 值才会导致次线性的查询时间。在附录的第 A 部分，我们给出了随机投影结果的一个版本，使用了比 Frankl 和 Maehara 更简单的证明。我们还考虑了随机投影方法对于 p 不等于 2 的 lp 范数的扩展。使用随机投影和 Proposition 2，我们得到了 Proposition 3 中描述的算法。不幸的是，高的预处理成本（其指数随 1/c增长）使得该算法在小 c 的情况下不实用。

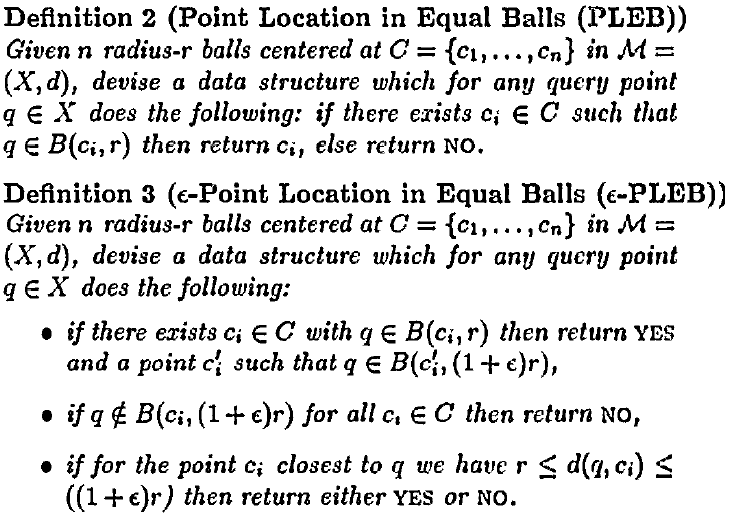
2.前言



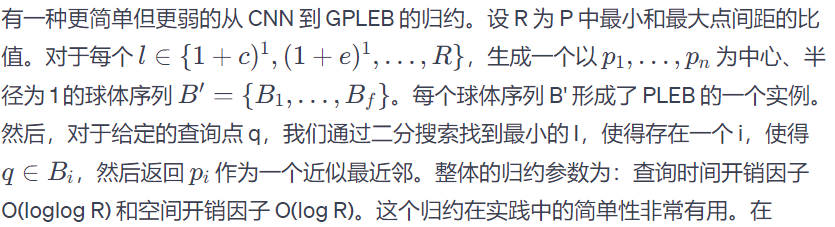


**3简化为等球中的点位置**

关键思想是将 e-NNS 简化为以下等球中的点定位问题。



注意到 PLEB（e-PLEB）可以被归约到 NNS（e-NNS），且具有相同的预处理和查询成本，如下所示：只需找到一个精确（e-近似）的最近邻，然后将其与查询点 q 的距离与 r 进行比较即可。本节的主要观点是要展示从 e-NNS 到 e-PLEB 的逆向归约，其预处理和查询成本只有很小的额外开销。这个归约依赖于一种称为环覆盖树的数据结构。这个结构利用了这样一个事实，对于任何点集 P，我们可以找到一个环分隔器或一个覆盖。任一结构都可以把P分解称集合S1,…,Sl这样对于所有的I,|Si|≤c|P|对于c<1,并且对于b＜1+1/log2n这种分解具有以下特性：在搜索 P 时，可以快速将搜索限制到集合 S 之一。



另一方面，当R很大时，O(logR)的空间开销不可接受；通常R可能是无界的。在最终的版本中，我们将表明，通过使用此方法的变体，存储可以减少到 O(n2 log n)，但这仍然没有给出所需的 O(l/c)dO(n) 界限。

